UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/705,668	11/02/2000	Darrell D. Boggs	042390.P9576	6260

7590 07/09/2004

Eric S Hyman
Blakely Sokoloff Taylor & Zafman LLP
12400 Wilshire Boulevard
7th Floor
Los Angeles, CA 90025

EXAMINER	
LI, AIMEE J	

ART UNIT	PAPER NUMBER
2183	7

DATE MAILED: 07/09/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/705,668	BOGGS ET AL.	
	Examiner	Art Unit	
	Aimee J Li	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 December 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-24* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-24* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-24 have been considered.

Specification

2. The disclosure is objected to because of the following informalities: Please correct n page 2, line 21 "base don" with --based on--. Please correct on page 8, line 6, "118" with --40--.

3. Appropriate correction is required.

Double Patenting

4. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970);and, *In re Thorington,* 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

5. A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

6. Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

7. Claims 1-9 and 12-21 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-19 of copending Application No. 09/705,678. Although the conflicting claims are not identical, they are not patentably distinct from each other because all the limitations in the above claims are found within the copending applications claims.

8. This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

9. Claims 1-19 of copending Application No. 09/705,678 contain every element of claims 1-9 and 12-21 of the instant application and as such anticipate claims 1-9 and 12-21of the instant application. Please see below for a table mapping the claim limitations of the instant application with the copending application.

Copending Application 09/705,678	Instant Application 09/705,668
A replay queue to receive a plurality of instructions (Claim 1);	A replay queue to receive a plurality of instructions (Claim 1);
An execution unit to execute the plurality of instructions (Claim 1);	An execution unit to execute the plurality of instructions (Claim 1);
A scheduler coupled between the replay queue and the execution unit to speculatively schedule instructions for execution based on data dependencies and expected latencies of said plurality of instructions (Claim 1);	A scheduler coupled between the replay queue and the execution unit to speculatively schedule instructions for execution (Claim 1);
A counter to count a number of times an instruction has one of executed and replayed (Claim 1),	To increment a counter for each of the plurality of instructions to reflect the number of times each of the plurality of instructions has been executed (Claim 1);
Wherein independent instructions and associated dependent instructions are executed if the counter is less than a predetermined value and if the counter exceeds the	To dispatch each instruction of the plurality of instructions to the execution unit either when the counter does not exceed a maximum number of replays or, if the counter for the

predetermined value the instruction is prevented from executing until data required by the instruction is available (Claim 1);	instruction exceeds the maximum number of replays, when the instruction is safe to execute (Claim 1);
A checker coupled to the execution unit to determine whether each instruction of the plurality of instructions has executed successfully, and coupled to the replay queue to dispatch to the replay queue each instruction that has not executed successfully (Claim 1).	A checker coupled to the execution unit to determine whether each instruction has executed successfully, and couples to the replay queue to communicate to the replay queue each instruction that has not executed successfully (Claim 1).
An allocator/renamer coupled to the replay queue to allocate and rename those of a plurality of resources needed by the instruction (Claims 2 and 13).	An allocator/renamer coupled to the replay queue to allocate and rename those of a plurality of resources needed by the instruction (Claims 2 and 15)
A front end coupled to the allocator/renamer to provide the plurality of instructions to the allocator/renamer (Claims 3 and 14).	A front end coupled to the allocator/renamer to provide the plurality of instructions to the allocator/renamer (Claims 3 and 16).
A retire unit to retire the plurality of instructions, coupled to the checker to receive those of the plurality of instructions that have executed successfully, and coupled to the allocator/renamer to communicate a de-allocate signal to the allocator/renamer (Claims 4 and	A retire unit to retire the plurality of instructions, coupled to the checker to receive those of the plurality of instructions that have executed successfully, and coupled to the allocator/renamer to communicate a de-allocate signal to the allocator/renamer (Claims 4 and

15).	17).
Wherein the retire unit is further coupled to the replay queue to communicate a retire signal when one of the plurality of instructions is retired such that the retired instruction and a plurality of associated data are removed from the replay queue (Claims 5 and 16).	Wherein the retire unit is further coupled to the replay queue to communicate a retire signal when one of the plurality of instructions is retired (Claims 5 and 18).
At least one cache system on a die of the processor (Claim 6);	At least one cache system on a die of the processor (Claim 6);
A plurality of external memory devices (Claim 6); and	A plurality of external memory devices (Claim 6); and
A memory request controller coupled to the execution unit to obtain a plurality of data from the at least one cache system and the plurality of external memory devices and to provide the plurality of data to the execution unit (Claim 6).	A memory request controller coupled to the execution unit to obtain a plurality of data from the at least one cache system and the plurality of external memory devices (Claim 6).
Wherein the at least one cache system comprises a first level cache system and a second level cache system (Claim 7).	Wherein the at least one cache system comprises a first level cache system and a second level cache system (Claim 7).
Wherein the external memory devices comprise at least one of a third level cache	Wherein the external memory devices comprise at least one of a third level cache

system, a main memory, and a disk memory (Claim 8).	system, a main memory, and a disk memory (Claim 8).
A staging queue coupled between the checker and the scheduler (Claim 9).	A staging queue coupled between the checker and the scheduler (Claim 9).
Wherein the checker comprises a scoreboard to maintain a status of a plurality of resources (Claim 10).	Wherein the checker comprises a scoreboard to maintain a status of a plurality of resources (Claim 12).
A replay queue to receive a plurality of instructions (Claim 11);	A replay queue to receive a plurality of instructions (Claim 13);
At least two execution units to execute the plurality of instructions (Claim 11);	At least two execution units to execute the plurality of instructions (Claim 13);
At least two schedulers coupled between the replay queue and the execution units to schedule instructions for execution based on data dependencies and instruction latencies (Claim 11);	At least two schedulers coupled between the replay queue and the execution units to schedule instructions for execution (Claim 13);
A counter to count a number of times an instruction has one of executed and replayed (Claim 11);	To increment a counter for each of the plurality of instructions to reflect the number of times each of the plurality of instructions has been executed (Claim 13);
Wherein independent instructions and associated dependent instructions are executed	To dispatch each instruction of the plurality of instructions to the execution unit either when

if the counter is less than a predetermined value and if the counter exceeds the predetermined value the instruction is prevented from executing until data required by the instruction is available (Claim 11);	the counter does not exceed a maximum number of replays or, if the counter for the instruction exceeds the maximum number of replays, when the instruction is safe to execute (Claim 13);
A checker coupled to the execution unit to determine whether each instruction has executed successfully, and couples to the replay queue to communicate to the replay queue each instruction that has not executed successfully (Claim 11).	A checker coupled to the execution unit to determine whether each instruction has executed successfully, and couples to the replay queue to communicate to the replay queue each instruction that has not executed successfully (Claim 13).
A plurality of memory devices coupled to the execution units such that the checker determines whether the instruction has executed successfully based on a plurality of information provided by the memory devices (Claim 12).	A plurality of memory devices coupled to the execution units such that the checker determines whether the instruction has executed successfully based on a plurality of information provided by the memory devices (Claim 14).
Receiving an instruction of a plurality of instructions (Claim 17);	Receiving an instruction of a plurality of instructions (Claim 19);
Placing the instruction in a queue with other instructions of the plurality of instructions (Claim 17);	Placing the instruction in a queue with other instructions of the plurality of instructions (Claim 19);

Speculatively re-ordering those of the plurality of instructions in a scheduler based on data dependencies and instruction latencies (Claim 17);	Speculatively re-ordering those of the plurality of instructions in a scheduler based on data dependencies and instruction latencies (Claim 19);
Dispatching one of the plurality of instructions to an execution unit to be executed (Claim 17)	Dispatching one of the plurality of instructions to an execution unit to be executed (Claim 19)
Counting a number of times an instruction has one of executed and replayed, wherein the instruction and associated dependent instructions are executed if the number of times the instruction has one of executed and re la ed is less than a predetermined value and if the number of times the instruction has one of executed and replayed exceeds the predetermined value the instruction is prevented from executing until data required by the instruction is available (Claim 17);	Either when a counter for the instruction does not exceed a maximum number of replays or, if the counter for the instruction exceed the maximum number of replays, when a required data for the instruction is available (Claim 19)
Executing the instruction (Claim 17);	Executing the instruction (Claim 19);
Determining whether the instruction executed successfully (Claim 17); and	Determining whether the instruction executed successfully (Claim 19); and
Routing the instruction and all associated dependent instructions back to the queue if the	Routing the instruction back to the queue if the instruction did not execute successfully (Claim

instruction did not execute successfully (Claim 17);	19).
Retiring the instruction if the instruction executed successfully and allowing the instruction's associated dependent instructions to execute (Claim 17),	Retiring the instruction if the instruction executed successfully (Claim 19);
Allocating those of a plurality of system resources used by the instruction being retired (Claim 18).	Allocating those of a plurality of system resources used by the instruction being retired (Claim 20).
De-allocating those of the plurality of system resources used by the instruction being retired (Claim 18); and	De-allocating those of the plurality of system resources used by the instruction being retired (Claim 21); and
Removing the instruction and a plurality of related data from the queue (Claim 18).	Removing the instruction and a plurality of related data from the queue (Claim 21).

Claim Rejections - 35 USC § 112

10. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

11. Claim 10 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claim 10 recites "a plurality of counters to maintain each of the plurality of

counters for each of the plurality of instructions" and it is unclear how a plurality of counters can maintain itself.

Claim Rejections - 35 USC § 103

12. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

13. Claims 1-4, 6-12, 19-20, and 22-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sager, U.S. Patent Number 5,966,544 (herein referred to as Sager) in view of Heath et al., U.S. Patent Number 3,603,934 (herein referred to as Heath).

14. Referring to claim 1, Sager has taught a processor comprising:

a. A replay queue to receive a plurality of instructions (Sager column 9, line 50 to column 10, line 6 and Figure 7);

b. An execution unit to execute the plurality of instructions (Sager column 8, lines 64-67 and Figure 7);

c. A scheduler coupled between the replay queue and the execution unit to speculatively schedule instructions for execution (Sager column 10, lines 7-33 and Figure 7);

d. A checker coupled to the execution unit to determine whether each instruction has executed successfully, and couples to the replay queue to communicate to the

replay queue each instruction that has not executed successfully (Sager column 9, lines 20-53 and Figure 7).

15. Sager has not explicitly taught to increment a counter for each of the plurality of instructions to reflect the number of times each of the plurality of instructions has been executed, and to dispatch each instruction of the plurality of instructions to the execution unit either when the counter does not exceed a maximum number of replays or, if the counter for the instruction exceeds the maximum number of replays, when the instruction is safe to execute. Heath has taught the general idea of when an instruction is erroneously executed, the instruction will be re-executed a predetermined number of times and when that number is exceeded, corrective measures have to be taken (Heath column 2, lines 1-14). A person of ordinary skill in the art would have recognized that by using the idea of Heath, a counter could be implemented in Sager so that a predetermined number of replays are attempted, thereby trying to solve the execution error through re-execution. However, when the predetermined number is exceeded, the re-execution of the erroneous instruction would be suspended so that other instructions, which would possibly execute correctly, would use the system resources instead of the continuously erroneous instruction. This would increase efficiency of the system in that a lot of time and resources are not wasted trying to repeatedly execute an error-causing instruction. If the error is not resolved after a certain amount of time, then its re-execution should be suspended and corrective measures should be taken to make the instruction safe to execute. This way, other instructions may be executed during the time that would have previously been spent executing the erroneous instruction. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention was made to incorporate the counter and error correction of Heath

in the device of Sager to execute other instructions instead of the erroneous instruction thereby increasing processor efficiency.

16. Referring to claim 2, Sager has taught an allocator/renamer coupled to the replay queue to allocate and rename those of a plurality of resources needed by the instruction (Sager column 8, lines 34-51 and Figure 7).

17. Referring to claim 3, Sager has taught a front end coupled to the allocator/renamer to provide the plurality of instructions to the allocator/renamer (Sager column 8, lines 29-33 and Figure 7).

18. Referring to claim 4, Sager has taught a retire unit to retire the plurality of instructions coupled to the checker to receive those of the plurality of instructions that have executed successfully, and coupled to the allocator/renamer to communicate a de-allocate signal to the allocator/renamer (Sager column 14, lines 12-20; Figure 7; and Figure 8).

19. Referring to claim 6, Sager has taught

a. At least one cache on a die of the processor (Sager columns 2-3, lines 66-7; column 7, lines 25-26; Figure 2; and Figure 7);

b. A plurality of external memory devices (Sager columns 2-3, lines 66-7; Figure 2 and Figure 6); and

c. A memory request controller coupled to the execution unit to obtain data from the at least one cache system and the plurality of external memory devices (Sager columns 8-9, lines 64-19 and Figure 7).

20. Referring to claim 7, Sager has taught at least one cache system comprises a first level cache system and a second level cache system (Sager columns 2-3, lines 66-7 and Figure 2).

21. Referring to claim 8, Sager has taught Sager has taught wherein the external memory devices comprise at least one of a third level cache system, a main memory, and a disk memory (Sager columns 2-3, lines 66-7 and Figure 2).

22. Referring to claim 9, Sager has taught a staging queue coupled between the checker and the scheduler (Sager column 9, lines 31-49 and Figure 7).

23. Referring to claims 10 and 11, Sager has not taught

a. Wherein the scheduler comprises a plurality of counters to maintain each of the plurality of counters for each of the plurality of instructions (Applicant's claim 10); and

b. Wherein the counter is one of a plurality of counters such that each counter of the plurality of counters is paired with one of the plurality of instructions (Applicant's claim 11).

24. Heath has taught the general idea of when an instruction is erroneously executed, the instruction will be re-executed a predetermined number of times and when that number is exceeded, corrective measures have to be taken (Heath column 2, lines 1-14). A person of ordinary skill in the art would have recognized that by using the idea of Heath, a counter could be implemented in Sager so that a predetermined number of replays are attempted, thereby trying to solve the execution error through re-execution. However, when the predetermined number is exceeded, the re-execution of the erroneous instruction would be suspended so that other instructions, which would possibly execute correctly, would use the system resources instead of the continuously erroneous instruction. This would increase efficiency of the system in that a lot of time and resources are not wasted trying to repeatedly execute an error-causing instruction. If

the error is not resolved after a certain amount of time, then its re-execution should be suspended and corrective measures should be taken to make the instruction safe to execute. This way, other instructions may be executed during the time that would have previously been spent executing the erroneous instruction. In regards to Heath, a one-to-one ratio of counter-to-instruction has been taught, meaning that when there are multiple instructions there will be an equal amount of multiple counters, since an individual counter counts the number of times an individual instruction is executed. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention was made to incorporate the counter and error correction of Heath in the device of Sager to execute other instructions instead of the erroneous instruction thereby increasing processor efficiency.

25. Referring to claim 12, Sager has taught wherein the checker comprises a scoreboard to maintain a status of a plurality of resources (Sager column 9, lines 36-43 and 50-53 and Figure 7). In regards to Sager, the checker checks the dependencies associated with the instruction and scoreboarding is a common technique used to do this.

26. Referring to claim 19, Sager has taught a method comprising:

a. Receiving an instruction of a plurality of instructions (Sager column 8, lines 32-33; column 9, lines 34-43; column 9, line 63 to column 10, line 6; and Figure 7);

b. Placing the instruction in a queue with other instructions of the plurality of instructions (Sager column 8, lines 32-33; columns 9-10, lines 63-6; and Figure 7);

c. Speculatively re-ordering those of the plurality of instructions in a scheduler based on data dependencies and instruction latencies (Sager column 8, lines 52-63);

d. Executing the instruction (Sager column 8, lines 64-67 and Figure 7);

e. Determining whether the instruction executed successfully (Sager column 9, lines 20-49 and Figure 7);

f. Routing the instruction back to the queue if the instruction did not execute successfully (Sage column 9, lines 20-53; column 10, lines 47-52; column 12, lines 4-8; and Figure 7); and

g. Retiring the instruction if the instruction executed successfully (Sager column 10, lines 52-56; column 12, lines 32-37; column 14, lines 12-20; Figure 7; and Figure 8).

27. Sager has not taught dispatching one of the plurality of instructions to an execution unit to be executed either when a counter for the instruction does not exceed a maximum number of replays or, if the counter for the instruction exceed the maximum number of replays, when a required data for the instruction is available. Heath has taught the general idea of when an instruction is erroneously executed, the instruction will be re-executed a predetermined number of times and when that number is exceeded, corrective measures have to be taken (Heath column 2, lines 1-14). A person of ordinary skill in the art would have recognized that by using the idea of Heath, a counter could be implemented in Sager so that a predetermined number of replays are attempted, thereby trying to solve the execution error through re-execution. However, when the predetermined number is exceeded, the re-execution of the erroneous instruction would be

suspended so that other instructions, which would possibly execute correctly, would use the system resources instead of the continuously erroneous instruction. This would increase efficiency of the system in that a lot of time and resources are not wasted trying to repeatedly execute an error-causing instruction. If the error is not resolved after a certain amount of time, then its re-execution should be suspended and corrective measures should be taken to make the instruction safe to execute. This way, other instructions may be executed during the time that would have previously been spent executing the erroneous instruction. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention was made to incorporate the counter and error correction of Heath in the device of Sager to execute other instructions instead of the erroneous instruction thereby increasing processor efficiency.

28. Referring to claim 20, Sager has taught allocating those of a plurality of system resources used by the instruction being retired (Sager column 2, line 66 to column 3, line 7; column 8, line 64 to column 9, line 19; column 13, lines 61-64; Figure 2; and Figure 7).

29. Referring to claims 22-24, Sager has not taught

a. Maintaining a plurality of counters, one each for each of the plurality of instructions in the scheduler such that the counters reflect the number of times the corresponding instruction has been executed (Applicant's claim 22);

b. Wherein each of the plurality of counters for the instruction is paired with each of the plurality of the instructions (Applicant's claim 23); and

c. Wherein the plurality of counters is stored in the scheduler (Applicant's claim 24).

30. Heath has taught the general idea of when an instruction is erroneously executed, the instruction will be re-executed a predetermined number of times and when that number is exceeded, corrective measures have to be taken (Heath column 2, lines 1-14). A person of ordinary skill in the art would have recognized that by using the idea of Heath, a counter could be implemented in Sager so that a predetermined number of replays are attempted, thereby trying to solve the execution error through re-execution. However, when the predetermined number is exceeded, the re-execution of the erroneous instruction would be suspended so that other instructions, which would possibly execute correctly, would use the system resources instead of the continuously erroneous instruction. This would increase efficiency of the system in that a lot of time and resources are not wasted trying to repeatedly execute an error-causing instruction. If the error is not resolved after a certain amount of time, then its re-execution should be suspended and corrective measures should be taken to make the instruction safe to execute. This way, other instructions may be executed during the time that would have previously been spent executing the erroneous instruction. In regards to Heath, a one-to-one ratio of counter-to-instruction has been taught, meaning that when there are multiple instructions there will be an equal amount of multiple counters, since an individual counter counts the number of times an individual instruction is executed. Also, it does not matter where the counters are stored, since, no matter their position, they are functionally equivalent (See *In re Japikse*, 181 F.2d 1019, 86 USPQ 70 (CCPA 1950)). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention was made to incorporate the counter and error correction of Heath in the device of Sager to execute other instructions instead of the erroneous instruction thereby increasing processor efficiency.

31. Claims 5 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sager in view of Heath, as applied to claims 4 and 20 above, and further in view of Baxter et al., U.S. Patent Number 5,944,818 (herein referred to as Baxter).

32. Referring to claim 5, Sager has not taught the retire unit is further coupled to the replay queue to communicate a retire signal when one of the plurality of instructions is retired. Baxter has taught the retire unit is further coupled to the replay queue to communicate a retire signal when one of the plurality of instructions is retired (Baxter Fig.2 and column 3, lines 43-55). In regards to Baxter, upon retirement, the corresponding instruction entry in the replay queue (MIQ) is discarded (via deallocation signal shown in Fig.2) since there is no longer a need to maintain the instruction. Likewise, when an instruction retires in Sager, there would be no need to maintain that instruction in the replay queue. Doing so would consume resources for no beneficial reason. A person of ordinary skill in the art at the time the invention was made would have recognized that the retirement signal of Baxter allows the queue to acknowledge when to discard data, thereby freeing space in the queue to store more data. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate the signal of Baxter in the device of Sager to free space in the queue by deallocating data no longer needed.

33. Referring to claim 21, Sager has taught de-allocating those of the plurality of system resources used by the instruction being retired (Sager column 14, lines 12-20; Figure 7; and Figure 8). In regards to Sager, it is inherent that when an instruction is retired, its resources (i.e. registers) are deallocated so that another instruction has the option to use them (as opposed to resources continuing to be allocated to an instruction which no longer requires them because that

instruction has completed). If these resources were not deallocated, then they would not be available to the processor and, consequently, inhibiting execution. Sager has not taught removing the instruction and a plurality of related data from the queue. Baxter has taught removing the instruction and a plurality of related data from the queue (Baxter Fig.2 and column 3, lines 43-55). In regards to Baxter, upon retirement, the corresponding instruction entry in the replay queue (MIQ) is discarded (via deallocation signal shown in Fig.2) since there is no longer a need to maintain the instruction. Likewise, when an instruction retires in Sager, there would be no need to maintain that instruction in the replay queue. Doing so would consume resources for no beneficial reason. A person of ordinary skill in the art at the time the invention was made would have recognized that removing elements from the queue, as taught by Baxter, frees space in the queue to store more data. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate the removal of elements from the queue, as taught by Baxter, in the device of Sager to free space in the queue by removing data no longer needed.

34. Claims 13-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sager, U.S. Patent Number 5,966,544 (herein referred to as Sager) in view of Heath et al., U.S. Patent Number 3,603,934 (herein referred to as Heath) and in further view of Johnson's Superscalar Microprocessor Design ©1990 (herein referred to as Johnson).

35. Referring to claim 13, Sager has taught

a. A replay queue to receive a plurality of instructions (Sager columns 9-10, lines 63-6 and Figure 7);

b. At least two execution Units to execute the plurality of instructions (Sager column 5, lines 33-35; column 8, lines 64-67; and Figure 7); and

a. A scheduler coupled between the replay queue and the execution units to schedule instructions for execution (Sager column8, lines 52-63; column 10, lines 27-33; and Figure 7); and

b. A checker coupled to the execution unit to determine whether each instruction has executed successfully, and couples to the replay queue to communicate to the replay queue each instruction that has not executed successfully (Sager column 9, lines 20-49 and Figure 7).

36. Sager has not explicitly taught to increment a counter for each of the plurality of instructions to reflect the number of times each of the plurality of instructions has been executed, and to dispatch each instruction of the plurality of instructions to the execution unit either when the counter does not exceed a maximum number of replays or, if the counter for the instruction exceeds the maximum number of replays, when the instruction is safe to execute. Heath has taught the general idea of when an instruction is erroneously executed, the instruction will be re-executed a predetermined number of times and when that number is exceeded, corrective measures have to be taken (Heath column 2, lines 1-14). A person of ordinary skill in the art would have recognized that by using the idea of Heath, a counter could be implemented in Sager so that a predetermined number of replays are attempted, thereby trying to solve the execution error through re-execution. However, when the predetermined number is exceeded, the re-execution of the erroneous instruction would be suspended so that other instructions, which would possibly execute correctly, would use the system resources instead of the continuously

erroneous instruction. This would increase efficiency of the system in that a lot of time and resources are not wasted trying to repeatedly execute an error-causing instruction. If the error is not resolved after a certain amount of time, then its re-execution should be suspended and corrective measures should be taken to make the instruction safe to execute. This way, other instructions may be executed during the time that would have previously been spent executing the erroneous instruction. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention was made to incorporate the counter and error correction of Heath in the device of Sager to execute other instructions instead of the erroneous instruction thereby increasing processor efficiency.

37. In addition, Sager has not taught at least two schedulers. However, Sager has taught that a scheduler may be a reservation station (Sager column 8, lines 52-63). Johnson has taught multiple reservation stations, which includes two (Johnson page 134, lines 1-22). A person of ordinary skill in the art at the time the invention was made, and as taught by Johnson, would have recognized that multiple reservations stations is simpler logically than with one large scheduler (reservation station) for multiple reasons (Johnson page 134, lines 1-22). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the multiple schedulers of Johnson in the device of Sager to simplify the scheduler logic.

38. Referring to claim 14, Sager has taught a plurality of memory devices coupled to the execution units such that the checker determines whether the instruction has executed successfully based on a plurality of information provided by the memory devices (Sager columns

2, line 66 to column 3, line 7; column 8, lines 29-32; column 8, line 64 to column 9, line 19; Figure 2; and Figure 7).

39. Referring to claim 15, Sager has taught an allocator/renamer coupled to the replay queue to allocate and rename those of a plurality of resources needed by the instruction (Sager column 8, lines 34-51 and Figure 7).

40. Referring to claim 16, Sager has taught a front end coupled to the allocator/renamer to provide the plurality of instructions to the allocator/renamer (Sager column 8, lines 29-33 and Figure 7).

41. Referring to claim 17, Sager has taught a retire unit to retire the plurality of instructions coupled to the checker to receive those of the plurality of instructions that have executed successfully, and coupled to the allocator/renamer to communicate a de-allocate signal to the allocator/renamer (Sager column 14, lines 12-20; Figure 7; and Figure 8).

42. Claim 18 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sager in view of Heath, as applied to claim 17 above, and further in view of Baxter et al., U.S. Patent Number 5,944,818 (herein referred to as Baxter). Sager has not taught the retire unit is further coupled to the replay queue to communicate a retire signal when one of the plurality of instructions is retired. Baxter has taught the retire unit is further coupled to the replay queue to communicate a retire signal when one of the plurality of instructions is retired (Baxter Fig.2 and column 3, lines 43-55). In regards to Baxter, upon retirement, the corresponding instruction entry in the replay queue (MIQ) is discarded (via deallocation signal shown in Fig.2) since there is no longer a need to maintain the instruction. Likewise, when an instruction retires in Sager, there would be no need to maintain that instruction in the replay queue. Doing so would consume resources for no

beneficial reason. A person of ordinary skill in the art at the time the invention was made would have recognized that the retirement signal of Baxter allows the queue to acknowledge when to discard data, thereby freeing space in the queue to store more data. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate the signal of Baxter in the device of Sager to free space in the queue by deallocating data no longer needed.

43. Claims 1-4, 6, 9-12, and 19-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Merchant et al., U.S. Patent Number 6,212,626 (herein referred to as Merchant) in view of Heath et al., U.S. Patent Number 3,603,934 (herein referred to as Heath).

44. Referring to claim 1, Merchant has taught a processor comprising:

a. A replay queue to receive a plurality of instructions (Merchant Figure 1, element 70 especially stages 84 and 85);

b. An execution unit to execute the plurality of instructions (Merchant Figure 1, element 58);

c. A scheduler coupled between the replay queue and the execution unit to speculatively schedule instructions for execution (Merchant column 2, lines 39-46 and Figure 1);

d. A checker coupled to the execution unit to determine whether each instruction has executed successfully, and couples to the replay queue to communicate to the replay queue each instruction that has not executed successfully (Merchant Figure 1, element 72).

45. Merchant has not explicitly taught to increment a counter for each of the plurality of instructions to reflect the number of times each of the plurality of instructions has been executed, and to dispatch each instruction of the plurality of instructions to the execution unit either when the counter does not exceed a maximum number of replays or, if the counter for the instruction exceeds the maximum number of replays, when the instruction is safe to execute. Heath has taught the general idea of when an instruction is erroneously executed, the instruction will be re-executed a predetermined number of times and when that number is exceeded, corrective measures have to be taken (Heath column 2, lines 1-14). A person of ordinary skill in the art would have recognized that by using the idea of Heath, a counter could be implemented in Merchant so that a predetermined number of replays are attempted, thereby trying to solve the execution error through re-execution. However, when the predetermined number is exceeded, the re-execution of the erroneous instruction would be suspended so that other instructions, which would possibly execute correctly, would use the system resources instead of the continuously erroneous instruction. This would increase efficiency of the system in that a lot of time and resources are not wasted trying to repeatedly execute an error-causing instruction. If the error is not resolved after a certain amount of time, then its re-execution should be suspended and corrective measures should be taken to make the instruction safe to execute. This way, other instructions may be executed during the time that would have previously been spent executing the erroneous instruction. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention was made to incorporate the counter and error correction of Heath in the device of Merchant to execute other instructions instead of the erroneous instruction thereby increasing processor efficiency.

46. Referring to claim 2, Merchant has taught an allocator/renamer coupled to the replay queue to allocate and rename those of a plurality of resources needed by the instruction (Figure 1, element 2). In regards to Merchant, scoreboarding deal with renaming and allocating.

47. Referring to claim 3, Merchant has taught a front end coupled to the allocator/renamer to provide the plurality of instructions to the allocator/renamer (Merchant Figure 1, element 52).

48. Referring to claim 4, Merchant has taught a retire unit to retire the plurality of instructions coupled to the checker to receive those of the plurality of instructions that have executed successfully, and coupled to the allocator/renamer to communicate a de-allocate signal to the allocator/renamer (Merchant column 3, lines 27-29 and Figure 1, element 62).

49. Referring to claim 6, Merchant has taught

a. At least one cache on a die of the processor (Merchant column 3, lines 53-54). In regards to Merchant, a cache miss can only occur if a cache system exists.

b. A plurality of external memory devices (Merchant column 2, lines 24-27). In regards to Merchant, the external memory devices are inherent since they store the programs and data.

c. A memory request controller coupled to the execution unit to obtain data from the at least one cache system and the plurality of external memory devices (Merchant column 2, lines 24-27 and Figure 1, element bus 98).

50. Referring to claim 9, Merchant has taught a staging queue coupled between the checker and the scheduler (Merchant Figure 1, elements 80-83).

51. Referring to claims 10 and 11, Merchant has not taught

a. Wherein the scheduler comprises a plurality of counters to maintain each of the plurality of counters for each of the plurality of instructions (Applicant's claim 10); and

b. Wherein the counter is one of a plurality of counters such that each counter of the plurality of counters is paired with one of the plurality of instructions (Applicant's claim 11).

52. Heath has taught the general idea of when an instruction is erroneously executed, the instruction will be re-executed a predetermined number of times and when that number is exceeded, corrective measures have to be taken (Heath column 2, lines 1-14). A person of ordinary skill in the art would have recognized that by using the idea of Heath, a counter could be implemented in Merchant so that a predetermined number of replays are attempted, thereby trying to solve the execution error through re-execution. However, when the predetermined number is exceeded, the re-execution of the erroneous instruction would be suspended so that other instructions, which would possibly execute correctly, would use the system resources instead of the continuously erroneous instruction. This would increase efficiency of the system in that a lot of time and resources are not wasted trying to repeatedly execute an error-causing instruction. If the error is not resolved after a certain amount of time, then its re-execution should be suspended and corrective measures should be taken to make the instruction safe to execute. This way, other instructions may be executed during the time that would have previously been spent executing the erroneous instruction. In regards to Heath, a one-to-one ratio of counter-to-instruction has been taught, meaning that when there are multiple instructions there will be an equal amount of multiple counters, since an individual counter counts the

number of times an individual instruction is executed. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention was made to incorporate the counter and error correction of Heath in the device of Merchant to execute other instructions instead of the erroneous instruction thereby increasing processor efficiency.

53. Referring to claim 12, Merchant has taught wherein the checker comprises a scoreboard to maintain a status of a plurality of resources (Merchant Figure 1, element 54 and Figure 2).

54. Referring to claim 19, Merchant has taught a method comprising:

a. Receiving an instruction of a plurality of instructions (Merchant Figure 1, elements 80-83);

b. Placing the instruction in a queue with other instructions of the plurality of instructions (Merchant Figure 1, elements 84-85);

c. Speculatively re-ordering those of the plurality of instructions in a scheduler based on data dependencies and instruction latencies (Merchant column 2, lines 15-17 and 38-53);

d. Executing the instruction (Merchant column 2, lines 62-66 and Figure 1);

e. Determining whether the instruction executed successfully (Merchant column 3, lines 46-48).

f. Routing the instruction back to the queue if the instruction did not execute successfully (Merchant column 3, lines 17-32; column 6, line 46 to column 7, line 10; Figures 6A-6E); and

g. Retiring the instruction if the instruction executed successfully (Merchant column 3, lines 23-27; column 7, lines 11-42; Figure 1, element 62; and Figures 6F-6H).

55. Merchant has not taught dispatching one of the plurality of instructions to an execution unit to be executed either when a counter for the instruction does not exceed a maximum number of replays or, if the counter for the instruction exceed the maximum number of replays, when a required data for the instruction is available. However, Merchant has taught dispatching one of the plurality of instruction to an execution unit (Merchant column 2, lines 62-65 and Figure 1). Heath has taught the general idea of when an instruction is erroneously executed, the instruction will be re-executed a predetermined number of times and when that number is exceeded, corrective measures have to be taken (Heath column 2, lines 1-14). A person of ordinary skill in the art would have recognized that by using the idea of Heath, a counter could be implemented in Merchant so that a predetermined number of replays are attempted, thereby trying to solve the execution error through re-execution. However, when the predetermined number is exceeded, the re-execution of the erroneous instruction would be suspended so that other instructions, which would possibly execute correctly, would use the system resources instead of the continuously erroneous instruction. This would increase efficiency of the system in that a lot of time and resources are not wasted trying to repeatedly execute an error-causing instruction. If the error is not resolved after a certain amount of time, then its re-execution should be suspended and corrective measures should be taken to make the instruction safe to execute. This way, other instructions may be executed during the time that would have previously been spent executing the erroneous instruction. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention was made to incorporate the counter and error correction of Heath in the device of Merchant to execute other instructions instead of the erroneous instruction thereby increasing processor efficiency.

56. Referring to claim 20, Merchant has taught allocating those of a plurality of system resources used by the instruction being retired (Merchant column 2, lines 43-44 and Figure 1, scoreboard).

57. Referring to claim 21, Merchant has taught

a. De-allocating those of the plurality of system resources used by the instruction being retired (Merchant column 3, lines 17-29).

b. Removing the instruction and a plurality of related data from the queue (Merchant column 3, lines 17-29). If an instruction is eligible for retirement, then there is no need to keep it in the queue, since it won't need to execute again. Therefore, it is inherent that the instruction along with its data will be removed. Otherwise, if instructions were not removed, once the replay queue fills up due to its finite storage space, it will stay full and no additional instructions would be able to be stored for replay purposes.

58. Referring to claims 22-24, Merchant has not taught

a. Maintaining a plurality of counters, one each for each of the plurality of instructions in the scheduler such that the counters reflect the number of times the corresponding instruction has been executed (Applicant's claim 22);

b. Wherein each of the plurality of counters for the instruction is paired with each of the plurality of the instructions (Applicant's claim 23); and

c. Wherein the plurality of counters is stored in the scheduler (Applicant's claim 24).

59. Heath has taught the general idea of when an instruction is erroneously executed, the instruction will be re-executed a predetermined number of times and when that number is exceeded, corrective measures have to be taken (Heath column 2, lines 1-14). A person of ordinary skill in the art would have recognized that by using the idea of Heath, a counter could be implemented in Merchant so that a predetermined number of replays are attempted, thereby trying to solve the execution error through re-execution. However, when the predetermined number is exceeded, the re-execution of the erroneous instruction would be suspended so that other instructions, which would possibly execute correctly, would use the system resources instead of the continuously erroneous instruction. This would increase efficiency of the system in that a lot of time and resources are not wasted trying to repeatedly execute an error-causing instruction. If the error is not resolved after a certain amount of time, then its re-execution should be suspended and corrective measures should be taken to make the instruction safe to execute. This way, other instructions may be executed during the time that would have previously been spent executing the erroneous instruction. In regards to Heath, a one-to-one ratio of counter-to-instruction has been taught, meaning that when there are multiple instructions there will be an equal amount of multiple counters, since an individual counter counts the number of times an individual instruction is executed. Also, it does not matter where the counters are stored, since, no matter their position, they are functionally equivalent (See *In re Japikse*, 181 F.2d 1019, 86 USPQ 70 (CCPA 1950)). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention was made to incorporate the counter and error correction of Heath in the device of Merchant to execute other instructions instead of the erroneous instruction thereby increasing processor efficiency.

60. Claim 5 is rejected under 35 U.S.C. 103(a) as being unpatentable over Merchant in view of Heath, as applied to claims 4 and 20 above, and further in view of Baxter et al., U.S. Patent Number 5,944,818 (herein referred to as Baxter).

61. Referring to claim 5, Merchant has not taught the retire unit is further coupled to the replay queue to communicate a retire signal when one of the plurality of instructions is retired. Baxter has taught the retire unit is further coupled to the replay queue to communicate a retire signal when one of the plurality of instructions is retired (Baxter Fig.2 and column 3, lines 43-55). In regards to Baxter, upon retirement, the corresponding instruction entry in the replay queue (MIQ) is discarded (via deallocation signal shown in Fig.2) since there is no longer a need to maintain the instruction. Likewise, when an instruction retires in Merchant, there would be no need to maintain that instruction in the replay queue. Doing so would consume resources for no beneficial reason. A person of ordinary skill in the art at the time the invention was made would have recognized that the retirement signal of Baxter allows the queue to acknowledge when to discard data, thereby freeing space in the queue to store more data. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate the signal of Baxter in the device of Merchant to free space in the queue by deallocating data no longer needed.

62. Claims 13-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Merchant et al., U.S. Patent Number 6,212,626 (herein referred to as Merchant) in view of Heath et al., U.S. Patent Number 3,603,934 (herein referred to as Heath) and in further view of Johnson's Superscalar Microprocessor Design ©1990 (herein referred to as Johnson).

63. Referring to claim 13, Merchant has taught

a. A replay queue to receive a plurality of instructions (Merchant Figure 1, element 70 especially stages 84 and 85);

b. At least two execution Units to execute the plurality of instructions (Merchant column 2, lines 65-66 and Figure 1, element 58); and

c. A scheduler coupled between the replay queue and the execution units to schedule instructions for execution (Merchant column 2, lines 39-46 and Figure 1); and

d. A checker coupled to the execution unit to determine whether each instruction has executed successfully, and couples to the replay queue to communicate to the replay queue each instruction that has not executed successfully (Merchant Figure 1, element 72).

64. Merchant has not explicitly taught to increment a counter for each of the plurality of instructions to reflect the number of times each of the plurality of instructions has been executed, and to dispatch each instruction of the plurality of instructions to the execution unit either when the counter does not exceed a maximum number of replays or, if the counter for the instruction exceeds the maximum number of replays, when the instruction is safe to execute. Heath has taught the general idea of when an instruction is erroneously executed, the instruction will be re-executed a predetermined number of times and when that number is exceeded, corrective measures have to be taken (Heath column 2, lines 1-14). A person of ordinary skill in the art would have recognized that by using the idea of Heath, a counter could be implemented in Merchant so that a predetermined number of replays are attempted, thereby trying to solve the execution error through re-execution. However, when the predetermined number is exceeded, the re-execution of the erroneous instruction would be suspended so that other instructions,

which would possibly execute correctly, would use the system resources instead of the continuously erroneous instruction. This would increase efficiency of the system in that a lot of time and resources are not wasted trying to repeatedly execute an error-causing instruction. If the error is not resolved after a certain amount of time, then its re-execution should be suspended and corrective measures should be taken to make the instruction safe to execute. This way, other instructions may be executed during the time that would have previously been spent executing the erroneous instruction. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention was made to incorporate the counter and error correction of Heath in the device of Merchant to execute other instructions instead of the erroneous instruction thereby increasing processor efficiency.

65. In addition, Merchant has not taught at least two schedulers. Johnson has taught multiple schedulers, also known as reservation stations, which includes two (Johnson page 134, lines 1-22). A person of ordinary skill in the art at the time the invention was made, and as taught by Johnson, would have recognized that multiple reservations stations is simpler logically than with one large scheduler (reservation station) for multiple reasons (Johnson page 134, lines 1-22). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the multiple schedulers of Johnson in the device of Merchant to simplify the scheduler logic.

66. Referring to claim 14, Merchant has taught a plurality of memory devices coupled to the execution units such that the checker determines whether the instruction has executed successfully based on a plurality of information provided by the memory devices (Merchant Figure 1, element 52).

67. Referring to claim 15, Merchant has taught an allocator/renamer coupled to the replay queue to allocate and rename those of a plurality of resources needed by the instruction (Figure 1, element 2). In regards to Merchant, scoreboarding deal with renaming and allocating.

68. Referring to claim 16, Merchant has taught a front end coupled to the allocator/renamer to provide the plurality of instructions to the allocator/renamer (Merchant Figure 1, element 52).

69. Referring to claim 17, Merchant has taught a retire unit to retire the plurality of instructions coupled to the checker to receive those of the plurality of instructions that have executed successfully, and coupled to the allocator/renamer to communicate a de-allocate signal to the allocator/renamer (Merchant column 3, lines 27-29 and Figure 1, element 62).

70. Claim 18 is rejected under 35 U.S.C. 103(a) as being unpatentable over Merchant in view of Heath, as applied to claim 17 above, and further in view of Baxter et al., U.S. Patent Number 5,944,818 (herein referred to as Baxter). Merchant has not taught the retire unit is further coupled to the replay queue to communicate a retire signal when one of the plurality of instructions is retired. Baxter has taught the retire unit is further coupled to the replay queue to communicate a retire signal when one of the plurality of instructions is retired (Baxter Fig.2 and column 3, lines 43-55). In regards to Baxter, upon retirement, the corresponding instruction entry in the replay queue (MIQ) is discarded (via deallocation signal shown in Fig.2) since there is no longer a need to maintain the instruction. Likewise, when an instruction retires in Merchant, there would be no need to maintain that instruction in the replay queue. Doing so would consume resources for no beneficial reason. A person of ordinary skill in the art at the time the invention was made would have recognized that the retirement signal of Baxter allows the queue to acknowledge when to discard data, thereby freeing space in the queue to store more

data. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate the signal of Baxter in the device of Merchant to free space in the queue by deallocating data no longer needed.

Response to Arguments

71. Examiner must maintain drawing objections regarding Figure 1, elements 128 and 180. No preliminary amendment has been entered into the case nor was any unentered paper filed within the file wrapper.

72. Examiner withdraws drawing objections regarding elements 1011 and 1001 in favor of amendments made to the specification.

73. Applicant's arguments, see pages 7-11 of Amendment A, filed 03 December 2003, with respect to the rejection(s)of claim(s) 1-24 under Sager in view of Akkary have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of the above.

74. Applicant's representative states that Sager and Akkary fall under 35 U.S.C. § 103(c). However, Sager and Akkary would qualify as 102(b) references not 102(e) references. Akkary was published 24 June 1999 and Sager was published 12 October 1999. The current application was filed 02 November 2000 with no priority dates indicated on or in the file wrapper. Therefore, Akkary and Sager were both published over a year prior to the effective filing date of the current application.

Conclusion

75. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aimee J Li whose telephone number is (703) 305-7596. The examiner can normally be reached on M-T 7:30am-5:00pm.

76. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

77. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AJL
Aimee J. Li
July 6, 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100